Network Working Group
Request for Comments: 3267
Category: Standards Track

J. Sjoberg
M. Westerlund
Ericsson
A. Lakaniemi
Nokia
Q. Xie
Motorola

June 2002

Real-Time Transport Protocol (RTP) Payload Format and File Storage Format for the Adaptive Multi-Rate (AMR) and Adaptive Multi-Rate Wideband (AMR-WB) Audio Codecs

Status of this Memo

This document specifies an Internet standards track protocol for the Internet community, and requests discussion and suggestions for improvements. Please refer to the current edition of the "Internet Official Protocol Standards" (STD 1) for the standardization state and status of this protocol. Distribution of this memo is unlimited.

Copyright Notice

Abstract

This document specifies a real-time transport protocol (RTP) payload format to be used for Adaptive Multi-Rate (AMR) and Adaptive Multi-Rate Wideband (AMR-WB) encoded speech signals. The payload format is designed to be able to interoperate with existing AMR and AMR-WB transport formats on non-IP networks. In addition, a file format is specified for transport of AMR and AMR-WB speech data in storage mode applications such as email. Two separate MIME type registrations are included, one for AMR and one for AMR-WB, specifying use of both the RTP payload format and the storage format.

Table of Contents

1. Introduction

This document specifies the payload format for packetization of AMR and AMR-WB encoded speech signals into the Real-time Transport Protocol (RTP) [8]. The payload format supports transmission of multiple channels, multiple frames per payload, the use of fast codec mode adaptation, robustness against packet loss and bit errors, and interoperation with existing AMR and AMR-WB transport formats on non-IP networks, as described in Section 3.

The payload format itself is specified in Section 4. A related file format is specified in Section 5 for transport of AMR and AMR-WB speech data in storage mode applications such as email. In Section 8, two separate MIME type registrations are provided, one for AMR and one for AMR-WB.

Even though this RTP payload format definition supports the transport of both AMR and AMR-WB speech, it is important to remember that AMR and AMR-WB are two different codecs and they are always handled as different payload types in RTP.

2. Conventions and Acronyms

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC2119 [5].

The following acronyms are used in this document:

```
3GPP   - the Third Generation Partnership Project
AMR    - Adaptive Multi-Rate Codec
AMR-WB - Adaptive Multi-Rate Wideband Codec
CMR    - Codec Mode Request
CN     - Comfort Noise
DTX    - Discontinuous Transmission
```

ETSI - European Telecommunications Standards Institute
FEC - Forward Error Correction
SCR - Source Controlled Rate Operation
SID - Silence Indicator (the frames containing only CN parameters)
VAD - Voice Activity Detection
UED - Unequal Error Detection
UEP - Unequal Error Protection

The term "frame-block" is used in this document to describe the time-synchronized set of speech frames in a multi-channel AMR or AMR-WB session. In particular, in an N-channel session, a frame-block will contain N speech frames, one from each of the channels, and all N speech frames represents exactly the same time period.

3. Background on AMR/AMR-WB and Design Principles

AMR and AMR-WB were originally designed for circuit-switched mobile radio systems. Due to their flexibility and robustness, they are also suitable for other real-time speech communication services over packet-switched networks such as the Internet.

Because of the flexibility of these codecs, the behavior in a particular application is controlled by several parameters that select options or specify the acceptable values for a variable. These options and variables are described in general terms at appropriate points in the text of this specification as parameters to be established through out-of-band means. In Section 8, all of the parameters are specified in the form of MIME subtype registrations for the AMR and AMR-WB encodings. The method used to signal these parameters at session setup or to arrange prior agreement of the participants is beyond the scope of this document; however, Section 8.3 provides a mapping of the parameters into the Session Description Protocol (SDP) [11] for those applications that use SDP.

3.1. The Adaptive Multi-Rate (AMR) Speech Codec

The AMR codecs was originally developed and standardized by the European Telecommunications Standards Institute (ETSI) for GSM cellular systems. It is now chosen by the Third Generation Partnership Project (3GPP) as the mandatory codec for third generation (3G) cellular systems [1].

The AMR codec is a multi-mode codec that supports 8 narrow band speech encoding modes with bit rates between 4.75 and 12.2 kbps. The sampling frequency used in AMR is 8000 Hz and the speech encoding is performed on 20 ms speech frames. Therefore, each encoded AMR speech frame represents 160 samples of the original speech.

Among the 8 AMR encoding modes, three are already separately adopted as standards of their own. Particularly, the 6.7 kbps mode is adopted as PDC-EFR [14], the 7.4 kbps mode as IS-641 codec in TDMA [13], and the 12.2 kbps mode as GSM-EFR [12].

3.2. The Adaptive Multi-Rate Wideband (AMR-WB) Speech Codec

The Adaptive Multi-Rate Wideband (AMR-WB) speech codec [3] was originally developed by 3GPP to be used in GSM and 3G cellular systems.

Similar to AMR, the AMR-WB codec is also a multi-mode speech codec. AMR-WB supports 9 wide band speech coding modes with respective bit rates ranging from 6.6 to 23.85 kbps. The sampling frequency used in AMR-WB is 16000 Hz and the speech processing is performed on 20 ms frames. This means that each AMR-WB encoded frame represents 320 speech samples.

3.3. Multi-rate Encoding and Mode Adaptation

The multi-rate encoding (i.e., multi-mode) capability of AMR and AMR-WB is designed for preserving high speech quality under a wide range of transmission conditions.

With AMR or AMR-WB, mobile radio systems are able to use available bandwidth as effectively as possible. E.g., in GSM it is possible to dynamically adjust the speech encoding rate during a session so as to continuously adapt to the varying transmission conditions by dividing the fixed overall bandwidth between speech data and error protective coding to enable best possible trade-off between speech compression rate and error tolerance. To perform mode adaptation, the decoder (speech receiver) needs to signal the encoder (speech sender) the new mode it prefers. This mode change signal is called Codec Mode Request or CMR.

Since in most sessions speech is sent in both directions between the two ends, the mode requests from the decoder at one end to the encoder at the other end are piggy-backed over the speech frames in the reverse direction. In other words, there is no out-of-band signaling needed for sending CMRs.

Every AMR or AMR-WB codec implementation is required to support all the respective speech coding modes defined by the codec and must be able to handle mode switching to any of the modes at any time. However, some transport systems may impose limitations in the number of modes supported and how often the mode can change due to bandwidth

limitations or other constraints. For this reason, the decoder is allowed to indicate its acceptance of a particular mode or a subset of the defined modes for the session using out-of-band means.

For example, the GSM radio link can only use a subset of at most four different modes in a given session. This subset can be any combination of the 8 AMR modes for an AMR session or any combination of the 9 AMR-WB modes for an AMR-WB session.

Moreover, for better interoperability with GSM through a gateway, the decoder is allowed to use out-of-band means to set the minimum number of frames between two mode changes and to limit the mode change among neighboring modes only.

Section 8 specifies a set of MIME parameters that may be used to signal these mode adaptation controls at session setup.

3.4. Voice Activity Detection and Discontinuous Transmission

Both codecs support voice activity detection (VAD) and generation of comfort noise (CN) parameters during silence periods. Hence, the codecs have the option to reduce the number of transmitted bits and packets during silence periods to a minimum. The operation of sending CN parameters at regular intervals during silence periods is usually called discontinuous transmission (DTX) or source controlled rate (SCR) operation. The AMR or AMR-WB frames containing CN parameters are called Silence Indicator (SID) frames. See more details about VAD and DTX functionality in [9] and [10].

3.5. Support for Multi-Channel Session

Both the RTP payload format and the storage format defined in this document support multi-channel audio content (e.g., a stereophonic speech session).

Although AMR and AMR-WB codecs themselves do not support encoding of multi-channel audio content into a single bit stream, they can be used to separately encode and decode each of the individual channels.

To transport (or store) the separately encoded multi-channel content, the speech frames for all channels that are framed and encoded for the same 20 ms periods are logically collected in a frame-block.

At the session setup, out-of-band signaling must be used to indicate the number of channels in the session and the order of the speech frames from different channels in each frame-block. When using SDP for signaling, the number of channels is specified in the rtpmap

attribute and the order of channels carried in each frame-block is implied by the number of channels as specified in Section 4.1 in [24].

3.6. Unequal Bit-error Detection and Protection

The speech bits encoded in each AMR or AMR-WB frame have different perceptual sensitivity to bit errors. This property has been exploited in cellular systems to achieve better voice quality by using unequal error protection and detection (UEP and UED) mechanisms.

The UEP/UED mechanisms focus the protection and detection of corrupted bits to the perceptually most sensitive bits in an AMR or AMR-WB frame. In particular, speech bits in an AMR or AMR-WB frame are divided into class A, B, and C, where bits in class A are most sensitive and bits in class C least sensitive (see Table 1 below for AMR and [4] for AMR-WB). A frame is only declared damaged if there are bit errors found in the most sensitive bits, i.e., the class A bits. On the other hand, it is acceptable to have some bit errors in the other bits, i.e., class B and C bits.

Index	Mode	Class A bits	total speech bits
0	AMR 4.75	42	95
1	AMR 5.15	49	103
2	AMR 5.9	55	118
3	AMR 6.7	58	134
4	AMR 7.4	61	148
5	AMR 7.95	75	159
6	AMR 10.2	65	204
7	AMR 12.2	81	244
8	AMR SID	39	39

Table 1. The number of class A bits for the AMR codec.

Moreover, a damaged frame is still useful for error concealment at the decoder since some of the less sensitive bits can still be used. This approach can improve the speech quality compared to discarding the damaged frame.

3.6.1. Applying UEP and UED in an IP Network

To take full advantage of the bit-error robustness of the AMR and AMR-WB codec, the RTP payload format is designed to facilitate UEP/UED in an IP network. It should be noted however that the utilization of UEP and UED discussed below is OPTIONAL.

UEP/UED in an IP network can be achieved by detecting bit errors in class A bits and tolerating bit errors in class B/C bits of the AMR or AMR-WB frame(s) in each RTP payload.

Today there exist some link layers that do not discard packets with bit errors, e.g., SLIP and some wireless links. With the Internet traffic pattern shifting towards a more multimedia-centric one, more link layers of such nature may emerge in the future. With transport layer support for partial checksums, for example those supported by UDP-Lite [15], bit error tolerant AMR and AMR-WB traffic could achieve better performance over these types of links.

There are at least two basic approaches for carrying AMR and AMR-WB traffic over bit error tolerant IP networks:

1) Utilizing a partial checksum to cover headers and the most important speech bits of the payload. It is recommended that at least all class A bits are covered by the checksum.

2) Utilizing a partial checksum to only cover headers, but a frame CRC to cover the class A bits of each speech frame in the RTP payload.

In either approach, at least part of the class B/C bits are left without error-check and thus bit error tolerance is achieved.

Note, it is still important that the network designer pay attention to the class B and C residual bit error rate. Though less sensitive to errors than class A bits, class B and C bits are not insignificant and undetected errors in these bits cause degradation in speech quality. An example of residual error rates considered acceptable for AMR in UMTS can be found in [20] and for AMR-WB in [21].

The application interface to the UEP/UED transport protocol (e.g., UDP-Lite) may not provide any control over the link error rate, especially in a gateway scenario. Therefore, it is incumbent upon the designer of a node with a link interface of this type to choose a residual bit error rate that is low enough to support applications such as AMR encoding when transmitting packets of a UEP/UED transport protocol.

Approach 1 is a bit efficient, flexible and simple way, but comes with two disadvantages, namely, a) bit errors in protected speech bits will cause the payload to be discarded, and b) when transporting multiple frames in a payload there is the possibility that a single bit error in protected bits will cause all the frames to be discarded.

These disadvantages can be avoided, if needed, with some overhead in the form of a frame-wise CRC (Approach 2). In problem a), the CRC makes it possible to detect bit errors in class A bits and use the frame for error concealment, which gives a small improvement in speech quality. For b), when transporting multiple frames in a payload, the CRCs remove the possibility that a single bit error in a class A bit will cause all the frames to be discarded. Avoiding that gives an improvement in speech quality when transporting multiple frames over links subject to bit errors.

The choice between the above two approaches must be made based on the available bandwidth, and desired tolerance to bit errors. Neither solution is appropriate to all cases. Section 8 defines parameters that may be used at session setup to select between these approaches.

3.7. Robustness against Packet Loss

The payload format supports several means, including forward error correction (FEC) and frame interleaving, to increase robustness against packet loss.

3.7.1. Use of Forward Error Correction (FEC)

The simple scheme of repetition of previously sent data is one way of achieving FEC. Another possible scheme which is more bandwidth efficient is to use payload external FEC, e.g., RFC2733 [19], which generates extra packets containing repair data. The whole payload can also be sorted in sensitivity order to support external FEC schemes using UEP. There is also a work in progress on a generic version of such a scheme [18] that can be applied to AMR or AMR-WB payload transport.

With AMR or AMR-WB, it is possible to use the multi-rate capability of the codec to send redundant copies of the same mode or of another mode, e.g., one with lower-bandwidth. We describe such a scheme next.

This involves the simple retransmission of previously transmitted frame-blocks together with the current frame-block(s). This is done by using a sliding window to group the speech frame-blocks to send in each payload. Figure 1 below shows us an example.

```
--+--------+--------+--------+--------+--------+--------+--------+--
  | f(n-2) | f(n-1) |  f(n)  | f(n+1) | f(n+2) | f(n+3) | f(n+4) |
--+--------+--------+--------+--------+--------+--------+--------+--

  <---- p(n-1) ---->
           <----- p(n) ----->
                    <---- p(n+1) ---->
                             <---- p(n+2) ---->
                                      <---- p(n+3) ---->
                                               <---- p(n+4) ---->
```

Figure 1: An example of redundant transmission.

In this example each frame-block is retransmitted one time in the following RTP payload packet. Here, f(n-2)..f(n+4) denotes a sequence of speech frame-blocks and p(n-1)..p(n+4) a sequence of payload packets.

The use of this approach does not require signaling at the session setup. In other words, the speech sender can choose to use this scheme without consulting the receiver. This is because a packet containing redundant frames will not look different from a packet with only new frames. The receiver may receive multiple copies or versions (encoded with different modes) of a frame for a certain timestamp if no packet is lost. If multiple versions of the same speech frame are received, it is recommended that the mode with the highest rate be used by the speech decoder.

This redundancy scheme provides the same functionality as the one described in RFC 2198 "RTP Payload for Redundant Audio Data" [24]. In most cases the mechanism in this payload format is more efficient and simpler than requiring both endpoints to support RFC 2198 in addition. There are two situations in which use of RFC 2198 is indicated: if the spread in time required between the primary and redundant encodings is larger than 5 frame times, the bandwidth overhead of RFC 2198 will be lower; or, if a non-AMR codec is desired for the redundant encoding, the AMR payload format won’t be able to carry it.

The sender is responsible for selecting an appropriate amount of redundancy based on feedback about the channel, e.g., in RTCP receiver reports. A sender should not base selection of FEC on the CMR, as this parameter most probably was set based on none-IP

information, e.g., radio link performance measures. The sender is also responsible for avoiding congestion, which may be exacerbated by redundancy (see Section 6 for more details).

3.7.2. Use of Frame Interleaving

To decrease protocol overhead, the payload design allows several speech frame-blocks be encapsulated into a single RTP packet. One of the drawbacks of such an approach is that in case of packet loss this means loss of several consecutive speech frame-blocks, which usually causes clearly audible distortion in the reconstructed speech. Interleaving of frame-blocks can improve the speech quality in such cases by distributing the consecutive losses into a series of single frame-block losses. However, interleaving and bundling several frame-blocks per payload will also increase end-to-end delay and is therefore not appropriate for all types of applications. Streaming applications will most likely be able to exploit interleaving to improve speech quality in lossy transmission conditions.

This payload design supports the use of frame interleaving as an option. For the encoder (speech sender) to use frame interleaving in its outbound RTP packets for a given session, the decoder (speech receiver) needs to indicate its support via out-of-band means (see Section 8).

3.8. Bandwidth Efficient or Octet-aligned Mode

For a given session, the payload format can be either bandwidth efficient or octet aligned, depending on the mode of operation that is established for the session via out-of-band means.

In the octet-aligned format, all the fields in a payload, including payload header, table of contents entries, and speech frames themselves, are individually aligned to octet boundaries to make implementations efficient. In the bandwidth efficient format only the full payload is octet aligned, so fewer padding bits are added.

> Note, octet alignment of a field or payload means that the last octet is padded with zeroes in the least significant bits to fill the octet. Also note that this padding is separate from padding indicated by the P bit in the RTP header.

Between the two operation modes, only the octet-aligned mode has the capability to use the robust sorting, interleaving, and frame CRC to make the speech transport robust to packet loss and bit errors.

3.9. AMR or AMR-WB Speech over IP scenarios

The primary scenario for this payload format is IP end-to-end between two terminals, as shown in Figure 2. This payload format is expected to be useful for both conversational and streaming services.

```
+----------+                          +----------+
|          |    IP/UDP/RTP/AMR or     |          |
| TERMINAL |<------------------------>| TERMINAL |
|          |    IP/UDP/RTP/AMR-WB     |          |
+----------+                          +----------+
```

Figure 2: IP terminal to IP terminal scenario

A conversational service puts requirements on the payload format. Low delay is one very important factor, i.e., few speech frame-blocks per payload packet. Low overhead is also required when the payload format traverses low bandwidth links, especially as the frequency of packets will be high. For low bandwidth links it also an advantage to support UED which allows a link provider to reduce delay and packet loss or to reduce the utilization of link resources.

Streaming service has less strict real-time requirements and therefore can use a larger number of frame-blocks per packet than conversational service. This reduces the overhead from IP, UDP, and RTP headers. However, including several frame-blocks per packet makes the transmission more vulnerable to packet loss, so interleaving may be used to reduce the effect packet loss will have on speech quality. A streaming server handling a large number of clients also needs a payload format that requires as few resources as possible when doing packetization. The octet-aligned and interleaving modes require the least amount of resources, while CRC, robust sorting, and bandwidth efficient modes have higher demands.

Another scenario occurs when AMR or AMR-WB encoded speech will be transmitted from a non-IP system (e.g., a GSM or 3GPP network) to an IP/UDP/RTP VoIP terminal, and/or vice versa, as depicted in Figure 3.

```
         AMR or AMR-WB
         over
         I.366.{2,3} or +------+                        +----------+
         3G Iu or       |      |   IP/UDP/RTP/AMR or    |          |
         <------------->|  GW  |<---------------------->| TERMINAL |
         GSM Abis       |      |   IP/UDP/RTP/AMR-WB    |          |
         etc.           +------+                        +----------+
                            |
          GSM/3GPP network  |           IP network
                            |
```

Figure 3: GW to VoIP terminal scenario

In such a case, it is likely that the AMR or AMR-WB frame is packetized in a different way in the non-IP network and will need to be re-packetized into RTP at the gateway. Also, speech frames from the non-IP network may come with some UEP/UED information (e.g., a frame quality indicator) that will need to be preserved and forwarded on to the decoder along with the speech bits. This is specified in Section 4.3.2.

AMR's capability to do fast mode switching is exploited in some non-IP networks to optimize speech quality. To preserve this functionality in scenarios including a gateway to an IP network, a codec mode request (CMR) field is needed. The gateway will be responsible for forwarding the CMR between the non-IP and IP parts in both directions. The IP terminal should follow the CMR forwarded by the gateway to optimize speech quality going to the non-IP decoder. The mode control algorithm in the gateway must accommodate the delay imposed by the IP network on the response to CMR by the IP terminal.

The IP terminal should not set the CMR (see Section 4.3.1), but the gateway can set the CMR value on frames going toward the encoder in the non-IP part to optimize speech quality from that encoder to the gateway. The gateway can alternatively set a lower CMR value, if desired, as one means to control congestion on the IP network.

A third likely scenario is that IP/UDP/RTP is used as transport between two non-IP systems, i.e., IP is originated and terminated in gateways on both sides of the IP transport, as illustrated in Figure 4 below.

```
AMR or AMR-WB                                       AMR or AMR-WB
over                                                over
I.366.{2,3} or +------+                     +------+ I.366.{2,3} or
3G Iu or       |      |  IP/UDP/RTP/AMR or  |      | 3G Iu or
<------------->|  GW  |<------------------->|  GW  |<------------->
GSM Abis       |      |  IP/UDP/RTP/AMR-WB  |      | GSM Abis
etc.           +------+                     +------+ etc.
                   |                            |
 GSM/3GPP network  |          IP network        |  GSM/3GPP network
                   |                            |
```

Figure 4: GW to GW scenario

This scenario requires the same mechanisms for preserving UED/UEP and CMR information as in the single gateway scenario. In addition, the CMR value may be set in packets received by the gateways on the IP network side. The gateway should forward to the non-IP side a CMR value that is the minimum of three values:

- the CMR value it receives on the IP side;
- the CMR value it calculates based on its reception quality on the non-IP side; and
- a CMR value it may choose for congestion control of transmission on the IP side.

The details of the control algorithm are left to the implementation.

4. AMR and AMR-WB RTP Payload Formats

The AMR and AMR-WB payload formats have identical structure, so they are specified together. The only differences are in the types of codec frames contained in the payload. The payload format consists of the RTP header, payload header and payload data.

4.1. RTP Header Usage

The format of the RTP header is specified in [8]. This payload format uses the fields of the header in a manner consistent with that specification.

The RTP timestamp corresponds to the sampling instant of the first sample encoded for the first frame-block in the packet. The timestamp clock frequency is the same as the sampling frequency, so the timestamp unit is in samples.

The duration of one speech frame-block is 20 ms for both AMR and AMR-WB. For AMR, the sampling frequency is 8 kHz, corresponding to 160 encoded speech samples per frame from each channel. For AMR-WB, the sampling frequency is 16 kHz, corresponding to 320 samples per frame from each channel. Thus, the timestamp is increased by 160 for AMR and 320 for AMR-WB for each consecutive frame-block.

A packet may contain multiple frame-blocks of encoded speech or comfort noise parameters. If interleaving is employed, the frame-blocks encapsulated into a payload are picked according to the interleaving rules as defined in Section 4.4.1. Otherwise, each packet covers a period of one or more contiguous 20 ms frame-block intervals. In case the data from all the channels for a particular frame-block in the period is missing, for example at a gateway from some other transport format, it is possible to indicate that no data is present for that frame-block rather than breaking a multi-frame-block packet into two, as explained in Section 4.3.2.

To allow for error resiliency through redundant transmission, the periods covered by multiple packets MAY overlap in time. A receiver MUST be prepared to receive any speech frame multiple times, either in exact duplicates, or in different AMR rate modes, or with data present in one packet and not present in another. If multiple versions of the same speech frame are received, it is RECOMMENDED that the mode with the highest rate be used by the speech decoder. A given frame MUST NOT be encoded as speech in one packet and comfort noise parameters in another.

The payload is always made an integral number of octets long by padding with zero bits if necessary. If additional padding is required to bring the payload length to a larger multiple of octets or for some other purpose, then the P bit in the RTP in the header may be set and padding appended as specified in [8].

The RTP header marker bit (M) SHALL be set to 1 if the first frame-block carried in the packet contains a speech frame which is the first in a talkspurt. For all other packets the marker bit SHALL be set to zero (M=0).

The assignment of an RTP payload type for this new packet format is outside the scope of this document, and will not be specified here. It is expected that the RTP profile under which this payload format is being used will assign a payload type for this encoding or specify that the payload type is to be bound dynamically.

4.2. Payload Structure

The complete payload consists of a payload header, a payload table of contents, and speech data representing one or more speech frame-blocks. The following diagram shows the general payload format layout:

```
+----------------+-------------------+----------------
| payload header | table of contents | speech data ...
+----------------+-------------------+----------------
```

Payloads containing more than one speech frame-block are called compound payloads.

The following sections describe the variations taken by the payload format depending on whether the AMR session is set up to use the bandwidth-efficient mode or octet-aligned mode and any of the OPTIONAL functions for robust sorting, interleaving, and frame CRCs. Implementations SHOULD support both bandwidth-efficient and octet-aligned operation to increase interoperability.

4.3. Bandwidth-Efficient Mode

4.3.1. The Payload Header

In bandwidth-efficient mode, the payload header simply consists of a 4 bit codec mode request:

```
 0 1 2 3
+-+-+-+-+
|  CMR  |
+-+-+-+-+
```

CMR (4 bits): Indicates a codec mode request sent to the speech encoder at the site of the receiver of this payload. The value of the CMR field is set to the frame type index of the corresponding speech mode being requested. The frame type index may be 0-7 for AMR, as defined in Table 1a in [2], or 0-8 for AMR-WB, as defined in Table 1a in [4]. CMR value 15 indicates that no mode request is present, and other values are for future use.

The mode request received in the CMR field is valid until the next CMR is received, i.e., a newly received CMR value overrides the previous one. Therefore, if a terminal continuously wishes to receive frames in the same mode X, it needs to set CMR=X for all its outbound payloads, and if a terminal has no preference in which mode to receive, it SHOULD set CMR=15 in all its outbound payloads.

If receiving a payload with a CMR value which is not a speech mode or NO_DATA, the CMR MUST be ignored by the receiver.

In a multi-channel session, CMR SHOULD be interpreted by the receiver of the payload as the desired encoding mode for all the channels in the session.

An IP end-point SHOULD NOT set the CMR based on packet losses or other congestion indications, for several reasons:

- The other end of the IP path may be a gateway to a non-IP network (such as a radio link) that needs to set the CMR field to optimize performance on that network.

- Congestion on the IP network is managed by the IP sender, in this case at the other end of the IP path. Feedback about congestion SHOULD be provided to that IP sender through RTCP or other means, and then the sender can choose to avoid congestion using the most appropriate mechanism. That may include adjusting the codec mode, but also includes adjusting the level of redundancy or number of frames per packet.

The encoder SHOULD follow a received mode request, but MAY change to a lower-numbered mode if it so chooses, for example to control congestion.

The CMR field MUST be set to 15 for packets sent to a multicast group. The encoder in the speech sender SHOULD ignore mode requests when sending speech to a multicast session but MAY use RTCP feedback information as a hint that a mode change is needed.

The codec mode selection MAY be restricted by a session parameter to a subset of the available modes. If so, the requested mode MUST be among the signalled subset (see Section 8).

4.3.2. The Payload Table of Contents

The table of contents (ToC) consists of a list of ToC entries, each representing a speech frame.

In bandwidth-efficient mode, a ToC entry takes the following format:

```
 0 1 2 3 4 5
+-+-+-+-+-+-+
|F|  FT   |Q|
+-+-+-+-+-+-+
```

F (1 bit): If set to 1, indicates that this frame is followed by another speech frame in this payload; if set to 0, indicates that this frame is the last frame in this payload.

FT (4 bits): Frame type index, indicating either the AMR or AMR-WB speech coding mode or comfort noise (SID) mode of the corresponding frame carried in this payload.

The value of FT is defined in Table 1a in [2] for AMR and in Table 1a in [4] for AMR-WB. FT=14 (SPEECH_LOST, only available for AMR-WB) and FT=15 (NO_DATA) are used to indicate frames that are either lost or not being transmitted in this payload, respectively.

NO_DATA (FT=15) frame could mean either that there is no data produced by the speech encoder for that frame or that no data for that frame is transmitted in the current payload (i.e., valid data for that frame could be sent in either an earlier or later packet).

If receiving a ToC entry with a FT value in the range 9-14 for AMR or 10-13 for AMR-WB the whole packet SHOULD be discarded. This is to avoid the loss of data synchronization in the depacketization process, which can result in a huge degradation in speech quality.

Note that packets containing only NO_DATA frames SHOULD NOT be transmitted. Also, frame-blocks containing only NO_DATA frames at the end of a packet SHOULD NOT be transmitted, except in the case of interleaving. The AMR SCR/DTX is described in [6] and AMR-WB SCR/DTX in [7].

The extra comfort noise frame types specified in table 1a in [2] (i.e., GSM-EFR CN, IS-641 CN, and PDC-EFR CN) MUST NOT be used in this payload format because the standardized AMR codec is only required to implement the general AMR SID frame type and not those that are native to the incorporated encodings.

Q (1 bit): Frame quality indicator. If set to 0, indicates the corresponding frame is severely damaged and the receiver should set the RX_TYPE (see [6]) to either SPEECH_BAD or SID_BAD depending on the frame type (FT).

The frame quality indicator is included for interoperability with the ATM payload format described in ITU-T I.366.2, the UMTS Iu interface [16], as well as other transport formats. The frame quality indicator enables damaged frames to be forwarded to the speech decoder for error concealment. This can improve the speech quality comparing to dropping the damaged frames. See Section 4.4.2.1 for more details.

For multi-channel sessions, the ToC entries of all frames from a frame-block are placed in the ToC in consecutive order as defined in Section 4.1 in [24]. When multiple frame-blocks are present in a packet in bandwidth-efficient mode, they will be placed in the packet in order of their creation time.

Therefore, with N channels and K speech frame-blocks in a packet, there MUST be N*K entries in the ToC, and the first N entries will be from the first frame-block, the second N entries will be from the second frame-block, and so on.

The following figure shows an example of a ToC of three entries in a single channel session using bandwidth efficient mode.

```
 0                   1
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|1|  FT   |Q|1|  FT   |Q|0|  FT   |Q|
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Below is an example of how the ToC entries will appear in the ToC of a packet carrying 3 consecutive frame-blocks in a session with two channels (L and R).

```
+----+----+----+----+----+----+
| 1L | 1R | 2L | 2R | 3L | 3R |
+----+----+----+----+----+----+
|<------->|<------->|<------->|
  Frame-    Frame-    Frame-
  Block 1   Block 2   Block 3
```

4.3.3. Speech Data

Speech data of a payload contains one or more speech frames or comfort noise frames, as described in the ToC of the payload.

> Note, for ToC entries with FT=14 or 15, there will be no corresponding speech frame present in the speech data.

Each speech frame represents 20 ms of speech encoded with the mode indicated in the FT field of the corresponding ToC entry. The length of the speech frame is implicitly defined by the mode indicated in the FT field. The order and numbering notation of the bits are as specified for Interface Format 1 (IF1) in [2] for AMR and [4] for AMR-WB. As specified there, the bits of speech frames have been rearranged in order of decreasing sensitivity, while the bits of comfort noise frames are in the order produced by the encoder. The resulting bit sequence for a frame of length K bits is denoted d(0), d(1), ..., d(K-1).

4.3.4. Algorithm for Forming the Payload

The complete RTP payload in bandwidth-efficient mode is formed by packing bits from the payload header, table of contents, and speech frames, in order as defined by their corresponding ToC entries in the ToC list, contiguously into octets beginning with the most significant bits of the fields and the octets.

To be precise, the four-bit payload header is packed into the first octet of the payload with bit 0 of the payload header in the most significant bit of the octet. The four most significant bits (numbered 0-3) of the first ToC entry are packed into the least significant bits of the octet, ending with bit 3 in the least significant bit. Packing continues in the second octet with bit 4 of the first ToC entry in the most significant bit of the octet. If more than one frame is contained in the payload, then packing continues with the second and successive ToC entries. Bit 0 of the first data frame follows immediately after the last ToC bit, proceeding through all the bits of the frame in numerical order. Bits from any successive frames follow contiguously in numerical order for each frame and in consecutive order of the frames.

If speech data is missing for one or more speech frame within the sequence, because of, for example, DTX, a ToC entry with FT set to NO_DATA SHALL be included in the ToC for each of the missing frames, but no data bits are included in the payload for the missing frame (see Section 4.3.5.2 for an example).

4.3.5 Payload Examples

4.3.5.1. Single Channel Payload Carrying a Single Frame

The following diagram shows a bandwidth-efficient AMR payload from a single channel session carrying a single speech frame-block.

In the payload, no specific mode is requested (CMR=15), the speech frame is not damaged at the IP origin (Q=1), and the coding mode is AMR 7.4 kbps (FT=4). The encoded speech bits, d(0) to d(147), are arranged in descending sensitivity order according to [2]. Finally, two zero bits are added to the end as padding to make the payload octet aligned.

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| CMR=15|0| FT=4  |1|d(0)                                       |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                     d(147)|P|P|
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

4.3.5.2. Single Channel Payload Carrying Multiple Frames

The following diagram shows a single channel, bandwidth efficient compound AMR-WB payload that contains four frames, of which one has no speech data. The first frame is a speech frame at 6.6 kbps mode (FT=0) that is composed of speech bits d(0) to d(131). The second frame is an AMR-WB SID frame (FT=9), consisting of bits g(0) to g(39). The third frame is NO_DATA frame and does not carry any speech information, it is represented in the payload by its ToC entry. The fourth frame in the payload is a speech frame at 8.85 kpbs mode (FT=1), it consists of speech bits h(0) to h(176).

As shown below, the payload carries a mode request for the encoder on the receiver’s side to change its future coding mode to AMR-WB 8.85 kbps (CMR=1). None of the frames is damaged at IP origin (Q=1). The encoded speech and SID bits, d(0) to d(131), g(0) to g(39) and h(0) to h(176), are arranged in the payload in descending sensitivity order according to [4]. (Note, no speech bits are present for the third frame). Finally, seven 0s are padded to the end to make the payload octet aligned.

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| CMR=1 |1| FT=0  |1|1| FT=9  |1|1| FT=15 |1|0| FT=1  |1|d(0)   |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                         d(131)|
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|g(0)                                                           |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|          g(39)|h(0)                                           |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                          h(176)|P|P|P|P|P|P|P|
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

4.3.5.3. Multi-Channel Payload Carrying Multiple Frames

The following diagram shows a two channel payload carrying 3 frame-blocks, i.e., the payload will contain 6 speech frames.

In the payload all speech frames contain the same mode 7.4 kbit/s (FT=4) and are not damaged at IP origin. The CMR is set to 15, i.e., no specific mode is requested. The two channels are defined as left (L) and right (R) in that order. The encoded speech bits is designated dXY(0).. dXY(K-1), where X = block number, Y = channel, and K is the number of speech bits for that mode. Exemplifying this, for frame-block 1 of the left channel the encoded bits are designated as d1L(0) to d1L(147).

```
    0                   1                   2                   3
    0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   | CMR=15|1|1L FT=4|1|1|1R FT=4|1|1|2L FT=4|1|1|2R FT=4|1|1|3L FT|
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |4|1|0|3R FT=4|1|d1L(0)                                         |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |                                                               |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |                                                               |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |                                                               |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |                                              d1L(147)|d1R(0) |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   : ...                                                           :
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |                      d1R(147)|d2L(0)                          |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   : ...                                                           :
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |d2L(147|d2R(0)                                                 |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   : ...                                                           :
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |                                      d2R(147)|d3L(0)          |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   : ...                                                           :
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |              d3L(147)|d3R(0)                                  |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   : ...                                                           :
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |                                                      d3R(147)|
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

4.4. Octet-aligned Mode

4.4.1. The Payload Header

In octet-aligned mode, the payload header consists of a 4 bit CMR, 4 reserved bits, and optionally, an 8 bit interleaving header, as shown below:

```
 0                   1
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5
+-+-+-+-+-+-+-+-+- - - - - - - -
|  CMR  |R|R|R|R|  ILL  |  ILP  |
+-+-+-+-+-+-+-+-+- - - - - - - -
```

CMR (4 bits): same as defined in section 4.3.1.

R: is a reserved bit that MUST be set to zero. All R bits MUST be ignored by the receiver.

ILL (4 bits, unsigned integer): This is an OPTIONAL field that is present only if interleaving is signalled out-of-band for the session. ILL=L indicates to the receiver that the interleaving length is L+1, in number of frame-blocks.

ILP (4 bits, unsigned integer): This is an OPTIONAL field that is present only if interleaving is signalled. ILP MUST take a value between 0 and ILL, inclusive, indicating the interleaving index for frame-blocks in this payload in the interleave group. If the value of ILP is found greater than ILL, the payload SHOULD be discarded.

ILL and ILP fields MUST be present in each packet in a session if interleaving is signalled for the session. Interleaving MUST be performed on a frame-block basis (i.e., NOT on a frame basis) in a multi-channel session.

The following example illustrates the arrangement of speech frame-blocks in an interleave group during an interleave session. Here we assume ILL=L for the interleave group that starts at speech frame-block n. We also assume that the first payload packet of the interleave group is s and the number of speech frame-blocks carried in each payload is N. Then we will have:

```
Payload s (the first packet of this interleave group):
  ILL=L, ILP=0,
  Carry frame-blocks: n, n+(L+1), n+2*(L+1), ..., n+(N-1)*(L+1)

   Payload s+1 (the second packet of this interleave group):
```

```
     ILL=L, ILP=1,
     frame-blocks: n+1, n+1+(L+1), n+1+2*(L+1), ..., n+1+(N-1)*(L+1)
       ...

   Payload s+L (the last packet of this interleave group):
     ILL=L, ILP=L,
     frame-blocks: n+L, n+L+(L+1), n+L+2*(L+1), ..., n+L+(N-1)*(L+1)
```

The next interleave group will start at frame-block n+N*(L+1).

There will be no interleaving effect unless the number of frame-blocks per packet (N) is at least 2. Moreover, the number of frame-blocks per payload (N) and the value of ILL MUST NOT be changed inside an interleave group. In other words, all payloads in an interleave group MUST have the same ILL and MUST contain the same number of speech frame-blocks.

The sender of the payload MUST only apply interleaving if the receiver has signalled its use through out-of-band means. Since interleaving will increase buffering requirements at the receiver, the receiver uses MIME parameter "interleaving=I" to set the maximum number of frame-blocks allowed in an interleaving group to I.

When performing interleaving the sender MUST use a proper number of frame-blocks per payload (N) and ILL so that the resulting size of an interleave group is less or equal to I, i.e., N*(L+1)<=I.

4.4.2. The Payload Table of Contents and Frame CRCs

The table of contents (ToC) in octet-aligned mode consists of a list of ToC entries where each entry corresponds to a speech frame carried in the payload and, optionally, a list of speech frame CRCs, i.e.,

```
   +---------------------+
   | list of ToC entries |
   +---------------------+
   | list of frame CRCs  | (optional)
    - - - - - - - - - - -
```

Note, for ToC entries with FT=14 or 15, there will be no corresponding speech frame or frame CRC present in the payload.

The list of ToC entries is organized in the same way as described for bandwidth-efficient mode in 4.3.2, with the following exception; when interleaving is used the frame-blocks in the ToC will almost never be placed consecutive in time. Instead, the presence and order of the frame-blocks in a packet will follow the pattern described in 4.4.1.

The following example shows the ToC of three consecutive packets, each carrying 3 frame-blocks, in an interleaved two-channel session. Here, the two channels are left (L) and right (R) with L coming before R, and the interleaving length is 3 (i.e., ILL=2). This makes the interleave group 9 frame-blocks large.

```
Packet #1
---------

ILL=2, ILP=0:
+----+----+----+----+----+----+
| 1L | 1R | 4L | 4R | 7L | 7R |
+----+----+----+----+----+----+
|<------->|<------->|<------->|
  Frame-    Frame-    Frame-
  Block 1   Block 4   Block 7

Packet #2
---------

ILL=2, ILP=1:
+----+----+----+----+----+----+
| 2L | 2R | 5L | 5R | 8L | 8R |
+----+----+----+----+----+----+
|<------->|<------->|<------->|
  Frame-    Frame-    Frame-
  Block 2   Block 5   Block 8

Packet #3
---------

ILL=2, ILP=2:
+----+----+----+----+----+----+
| 3L | 3R | 6L | 6R | 9L | 9R |
+----+----+----+----+----+----+
|<------->|<------->|<------->|
  Frame-    Frame-    Frame-
  Block 3   Block 6   Block 9
```

A ToC entry takes the following format in octet-aligned mode:

```
 0 1 2 3 4 5 6 7
+-+-+-+-+-+-+-+-+
|F|  FT   |Q|P|P|
+-+-+-+-+-+-+-+-+
```

F (1 bit): see definition in Section 4.3.2.

FT (4 bits unsigned integer): see definition in Section 4.3.2.

Q (1 bit): see definition in Section 4.3.2.

P bits: padding bits, MUST be set to zero.

The list of CRCs is OPTIONAL. It only exists if the use of CRC is signalled out-of-band for the session. When present, each CRC in the list is 8 bit long and corresponds to a speech frame (NOT a frame-block) carried in the payload. Calculation and use of the CRC is specified in the next section.

4.4.2.1. Use of Frame CRC for UED over IP

The general concept of UED/UEP over IP is discussed in Section 3.6. This section provides more details on how to use the frame CRC in the octet-aligned payload header together with a partial transport layer checksum to achieve UED.

To achieve UED, one SHOULD use a transport layer checksum, for example, the one defined in UDP-Lite [15], to protect the RTP header, payload header, and table of contents bits in a payload. The frame CRC, when used, MUST be calculated only over all class A bits in the frame. Class B and C bits in the frame MUST NOT be included in the CRC calculation and SHOULD NOT be covered by the transport checksum.

> Note, the number of class A bits for various coding modes in AMR codec is specified as informative in [2] and is therefore copied into Table 1 in Section 3.6 to make it normative for this payload format. The number of class A bits for various coding modes in AMR-WB codec is specified as normative in table 2 in [4], and the SID frame (FT=9) has 40 class A bits. These definitions of class A bits MUST be used for this payload format.

Packets SHOULD be discarded if the transport layer checksum detects errors.

The receiver of the payload SHOULD examine the data integrity of the received class A bits by re-calculating the CRC over the received class A bits and comparing the result to the value found in the received payload header. If the two values mismatch, the receiver SHALL consider the class A bits in the receiver frame damaged and MUST clear the Q flag of the frame (i.e., set it to 0). This will subsequently cause the frame to be marked as SPEECH_BAD, if the FT of the frame is 0..7 for AMR or 0..8 for AMR-WB, or SID_BAD if the FT of the frame is 8 for AMR or 9 for AMR-WB, before it is passed to the speech decoder. See [6] and [7] more details.

The following example shows an octet-aligned ToC with a CRC list for a payload containing 3 speech frames from a single channel session (assuming none of the FTs is equal to 14 or 15):

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|1|  FT#1 |Q|P|P|1|  FT#2 |Q|P|P|0|  FT#3 |Q|P|P|     CRC#1     |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|     CRC#2     |     CRC#3     |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Each of the CRC’s takes 8 bits

```
  0   1   2   3   4   5   6   7
+---+---+---+---+---+---+---+---+
| c0| c1| c2| c3| c4| c5| c6| c7|
+---+---+---+---+---+---+---+---+
```

and is calculated by the cyclic generator polynomial,

```
  C(x) = 1 + x^2 + x^3 + x^4 + x^8
```

where ^ is the exponentiation operator.

In binary form the polynomial has the following form: 101110001 (MSB..LSB).

The actual calculation of the CRC is made as follows: First, an 8-bit CRC register is reset to zero: 00000000. For each bit over which the CRC shall be calculated, an XOR operation is made between the rightmost bit of the CRC register and the bit. The CRC register is then right shifted one step (inputting a "0" as the leftmost bit). If the result of the XOR operation mentioned above is a "1" "10111000" is then bit-wise XOR-ed into the CRC register. This operation is repeated for each bit that the CRC should cover. In this case, the first bit would be d(0) for the speech frame for which the CRC should cover. When the last bit (e.g., d(54) for AMR 5.9 according to Table 1 in Section 3.6) have been used in this CRC calculation, the contents in CRC register should simply be copied to the corresponding field in the list of CRC’s.

Fast calculation of the CRC on a general-purpose CPU is possible using a table-driven algorithm.

4.4.3. Speech Data

In octet-aligned mode, speech data is carried in a similar way to that in the bandwidth-efficient mode as discussed in Section 4.3.3, with the following exceptions:

- The last octet of each speech frame MUST be padded with zeroes at the end if not all bits in the octet are used. In other words, each speech frame MUST be octet-aligned.

- When multiple speech frames are present in the speech data (i.e., compound payload), the speech frames can be arranged either one whole frame after another as usual, or with the octets of all frames interleaved together at the octet level. Since the bits within each frame are ordered with the most error-sensitive bits first, interleaving the octets collects those sensitive bits from all frames to be nearer the beginning of the packet. This is called "robust sorting order" which allows the application of UED (such as UDP-Lite [15]) or UEP (such as the ULP [18]) mechanisms to the payload data. The details of assembling the payload are given in the next section.

The use of robust sorting order for a session MUST be agreed via out-of-band means. Section 8 specifies a MIME parameter for this purpose.

Note, robust sorting order MUST only be performed on the frame level and thus is independent of interleaving which is at the frame-block level, as described in Section 4.4.1. In other words, robust sorting can be applied to either non-interleaved or interleaved sessions.

4.4.4. Methods for Forming the Payload

Two different packetization methods, namely normal order and robust sorting order, exist for forming a payload in octet-aligned mode. In both cases, the payload header and table of contents are packed into the payload the same way; the difference is in the packing of the speech frames.

The payload begins with the payload header of one octet or two if frame interleaving is selected. The payload header is followed by the table of contents consisting of a list of one-octet ToC entries. If frame CRCs are to be included, they follow the table of contents with one 8-bit CRC filling each octet. Note that if a given frame has a ToC entry with FT=14 or 15, there will be no CRC present.

The speech data follows the table of contents, or the CRCs if present. For packetization in the normal order, all of the octets comprising a speech frame are appended to the payload as a unit. The speech frames are packed in the same order as their corresponding ToC entries are arranged in the ToC list, with the exception that if a given frame has a ToC entry with FT=14 or 15, there will be no data octets present for that frame.

For packetization in robust sorting order, the octets of all speech frames are interleaved together at the octet level. That is, the data portion of the payload begins with the first octet of the first frame, followed by the first octet of the second frame, then the first octet of the third frame, and so on. After the first octet of the last frame has been appended, the cycle repeats with the second octet of each frame. The process continues for as many octets as are present in the longest frame. If the frames are not all the same octet length, a shorter frame is skipped once all octets in it have been appended. The order of the frames in the cycle will be sequential if frame interleaving is not in use, or according to the interleave pattern specified in the payload header if frame interleaving is in use. Note that if a given frame has a ToC entry with FT=14 or 15, there will be no data octets present for that frame so that frame is skipped in the robust sorting cycle.

The UED and/or UEP is RECOMMENDED to cover at least the RTP header, payload header, table of contents, and class A bits of a sorted payload. Exactly how many octets need to be covered depends on the network and application. If CRCs are used together with robust sorting, only the RTP header, the payload header, and the ToC SHOULD be covered by UED/UEP. The means to communicate to other layers performing UED/UEP the number of octets to be covered is beyond the scope of this specification.

4.4.5. Payload Examples

4.4.5.1. Basic Single Channel Payload Carrying Multiple Frames

The following diagram shows an octet aligned payload from a single channel session that carries two AMR frames of 7.95 kbps coding mode (FT=5). In the payload, a codec mode request is sent (CMR=6), requesting the encoder at the receiver's side to use AMR 10.2 kbps coding mode. No frame CRC, interleaving, or robust-sorting is in use.

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| CMR=6 |R|R|R|R|1|FT#1=5 |Q|P|P|0|FT#2=5 |Q|P|P|   f1(0..7)    |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|   f1(8..15)   |  f1(16..23)   |  ....                         |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
: ...                                                           :
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                          ...   |f1(152..158) |P|   f2(0..7)    |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|   f2(8..15)   |  f2(16..23)   |  ....                         |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
: ...                                                           :
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                          ...   |f2(152..158) |P|
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Note, in above example the last octet in both speech frames is padded with one 0 to make it octet-aligned.

4.4.5.2. Two Channel Payload with CRC, Interleaving, and Robust-sorting

This example shows an octet aligned payload from a two channel session. Two frame-blocks, each containing 2 speech frames of 7.95 kbps coding mode (FT=5), are carried in this payload,

The two channels are left (L) and right (R) with L coming before R. In the payload, a codec mode request is also sent (CMR=6), requesting the encoder at the receiver's side to use AMR 10.2 kbps coding mode.

Moreover, frame CRC and frame-block interleaving are both enabled for the session. The interleaving length is 2 (ILL=1) and this payload is the first one in an interleave group (ILP=0).

The first two frames in the payload are the L and R channel speech frames of frame-block #1, consisting of bits f1L(0..158) and

f1R(0..158), respectively. The next two frames are the L and R channel frames of frame-block #3, consisting of bits f3L(0..158) and f3R(0..158), respectively, due to interleaving. For each of the four speech frames a CRC is calculated as CRC1L(0..7), CRC1R(0..7), CRC3L(0..7), and CRC3R(0..7), respectively. Finally, the payload is robust sorted.

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| CMR=6 |R|R|R|R| ILL=1 | ILP=0 |1|FT#1L=5|Q|P|P|1|FT#1R=5|Q|P|P|
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|1|FT#3L=5|Q|P|P|0|FT#3R=5|Q|P|P|     CRC1L     |     CRC1R     |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|     CRC3L     |     CRC3R     |   f1L(0..7)   |   f1R(0..7)   |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|   f3L(0..7)   |   f3R(0..7)   |  f1L(8..15)   |  f1R(8..15)   |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|  f3L(8..15)   |  f3R(8..15)   |  f1L(16..23)  |  f1R(16..23)  |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
: ...                                                           :
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| f3L(144..151) | f3R(144..151) |f1L(152..158)|P|f1R(152..158)|P|
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|f3L(152..158)|P|f3R(152..158)|P|
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Note, in above example the last octet in all the four speech frames is padded with one zero bit to make it octet-aligned.

4.5. Implementation Considerations

An application implementing this payload format MUST understand all the payload parameters in the out-of-band signaling used. For example, if an application uses SDP, all the SDP and MIME parameters in this document MUST be understood. This requirement ensures that an implementation always can decide if it is capable or not of communicating.

No operation mode of the payload format is mandatory to implement. The requirements of the application using the payload format should be used to determine what to implement. To achieve basic interoperability an implementation SHOULD at least implement both bandwidth-efficient and octet-aligned mode for single channel. The other operations mode: interleaving, robust sorting, frame-wise CRC in both single and multi-channel is OPTIONAL to implement.

5. AMR and AMR-WB Storage Format

The storage format is used for storing AMR or AMR-WB speech frames in a file or as an e-mail attachment. Multiple channel content is supported.

In general, an AMR or AMR-WB file has the following structure:

```
+------------------+
| Header           |
+------------------+
| Speech frame 1   |
+------------------+
: ...              :
+------------------+
| Speech frame n   |
+------------------+
```

Note, to preserve interoperability with already deployed implementations, single channel content uses a file header format different from that of multi-channel content.

5.1. Single channel Header

A single channel AMR or AMR-WB file header contains only a magic number and different magic numbers are defined to distinguish AMR from AMR-WB.

The magic number for single channel AMR files MUST consist of ASCII character string:

```
"#!AMR\n"
(or 0x2321414d520a in hexadecimal).
```

The magic number for single channel AMR-WB files MUST consist of ASCII character string:

```
"#!AMR-WB\n"
(or 0x2321414d522d57420a in hexadecimal).
```

Note, the "\n" is an important part of the magic numbers and MUST be included in the comparison, since, otherwise, the single channel magic numbers above will become indistinguishable from those of the multi-channel files defined in the next section.

5.2. Multi-channel Header

The multi-channel header consists of a magic number followed by a 32 bit channel description field, giving the multi-channel header the following structure:

```
+------------------+
| magic number     |
+------------------+
| chan-desc field  |
+------------------+
```

The magic number for multi-channel AMR files MUST consist of the ASCII character string:

"#!AMR_MC1.0\n"
(or 0x2321414d525F4D43312E300a in hexadecimal).

The magic number for multi-channel AMR-WB files MUST consist of the ASCII character string:

"#!AMR-WB_MC1.0\n"
(or 0x2321414d522d57425F4D43312E300a in hexadecimal).

The version number in the magic numbers refers to the version of the file format.

The 32 bit channel description field is defined as:

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|      Reserved bits                                    | CHAN  |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Reserved bits: MUST be set to 0 when written, and a reader MUST ignore them.

CHAN (4 bit unsigned integer): Indicates the number of audio channels contained in this storage file. The valid values and the order of the channels within a frame block are specified in Section 4.1 in [24].

5.3. Speech Frames

After the file header, speech frame-blocks consecutive in time are stored in the file. Each frame-block contains a number of octet-aligned speech frames equal to the number of channels, and stored in increasing order, starting with channel 1.

Each stored speech frame starts with a one octet frame header with the following format:

```
 0 1 2 3 4 5 6 7
+-+-+-+-+-+-+-+-+
|P|  FT   |Q|P|P|
+-+-+-+-+-+-+-+-+
```

The FT field and the Q bit are defined in the same way as in Section 4.1.2. The P bits are padding and MUST be set to 0.

Following this one octet header come the speech bits as defined in 4.3.3. The last octet of each frame is padded with zeroes, if needed, to achieve octet alignment.

The following example shows an AMR frame in 5.9 kbit coding mode (with 118 speech bits) in the storage format.

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|P| FT=2  |Q|P|P|                                               |
+-+-+-+-+-+-+-+-+                                               +
|                                                               |
+          Speech bits for frame-block n, channel k             +
|                                                               |
+                                                           +-+-+
|                                                           |P|P|
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Frame-blocks or speech frames lost in transmission and non-received frame-blocks between SID updates during non-speech periods MUST be stored as NO_DATA frames (frame type 15, as defined in [2] and [4]) or SPEECH_LOST (frame type 14, only available for AMR-WB) in complete frame-blocks to keep synchronization with the original media.

6. Congestion Control

The general congestion control considerations for transporting RTP data apply to AMR or AMR-WB speech over RTP as well. However, the multi-rate capability of AMR and AMR-WB speech coding may provide an advantage over other payload formats for controlling congestion since the bandwidth demand can be adjusted by selecting a different coding mode.

Another parameter that may impact the bandwidth demand for AMR and AMR-WB is the number of frame-blocks that are encapsulated in each RTP payload. Packing more frame-blocks in each RTP payload can reduce the number of packets sent and hence the overhead from IP/UDP/RTP headers, at the expense of increased delay.

If forward error correction (FEC) is used to combat packet loss, the amount of redundancy added by FEC will need to be regulated so that the use of FEC itself does not cause a congestion problem.

It is RECOMMENDED that AMR or AMR-WB applications using this payload format employ congestion control. The actual mechanism for congestion control is not specified but should be suitable for real-time flows, e.g., "Equation-Based Congestion Control for Unicast Applications" [17].

7. Security Considerations

RTP packets using the payload format defined in this specification are subject to the general security considerations discussed in [8].

As this format transports encoded speech, the main security issues include confidentiality and authentication of the speech itself. The payload format itself does not have any built-in security mechanisms. External mechanisms, such as SRTP [22], MAY be used.

This payload format does not exhibit any significant non-uniformity in the receiver side computational complexity for packet processing and thus is unlikely to pose a denial-of-service threat due to the receipt of pathological data.

7.1. Confidentiality

To achieve confidentiality of the encoded AMR or AMR-WB speech, all speech data bits will need to be encrypted. There is less a need to encrypt the payload header or the table of contents due to 1) that they only carry information about the requested speech mode, frame type, and frame quality, and 2) that this information could be useful to some third party, e.g., quality monitoring.

As long as the AMR or AMR-WB payload is only packed and unpacked at either end, encryption may be performed after packet encapsulation so that there is no conflict between the two operations.

Interleaving may affect encryption. Depending on the encryption scheme used, there may be restrictions on, for example, the time when keys can be changed. Specifically, the key change may need to occur at the boundary between interleave groups.

The type of encryption method used may impact the error robustness of the payload data. The error robustness may be severely reduced when the data is encrypted unless an encryption method without error-propagation is used, e.g., a stream cipher. Therefore, UED/UEP based on robust sorting may be difficult to apply when the payload data is encrypted.

7.2. Authentication

To authenticate the sender of the speech, an external mechanism has to be used. It is RECOMMENDED that such a mechanism protect all the speech data bits. Note that the use of UED/UEP may be difficult to combine with authentication because any bit errors will cause authentication to fail.

Data tampering by a man-in-the-middle attacker could result in erroneous depacketization/decoding that could lower the speech quality. Tampering with the CMR field may result in speech in a different quality than desired.

To prevent a man-in-the-middle attacker from tampering with the payload packets, some additional information besides the speech bits SHOULD be protected. This may include the payload header, ToC, frame CRCs, RTP timestamp, RTP sequence number, and the RTP marker bit.

7.3. Decoding Validation

When processing a received payload packet, if the receiver finds that the calculated payload length, based on the information of the session and the values found in the payload header fields, does not match the size of the received packet, the receiver SHOULD discard the packet. This is because decoding a packet that has errors in its length field could severely degrade the speech quality.

8. Payload Format Parameters

This section defines the parameters that may be used to select optional features of the AMR and AMR-WB payload formats. The parameters are defined here as part of the MIME subtype registrations

for the AMR and AMR-WB speech codecs. A mapping of the parameters into the Session Description Protocol (SDP) [11] is also provided for those applications that use SDP. Equivalent parameters could be defined elsewhere for use with control protocols that do not use MIME or SDP.

Two separate MIME registrations are made, one for AMR and one for AMR-WB, because they are distinct encodings that must be distinguished by the MIME subtype.

The data format and parameters are specified for both real-time transport in RTP and for storage type applications such as e-mail attachments.

8.1. AMR MIME Registration

The MIME subtype for the Adaptive Multi-Rate (AMR) codec is allocated from the IETF tree since AMR is expected to be a widely used speech codec in general VoIP applications. This MIME registration covers both real-time transfer via RTP and non-real-time transfers via stored files.

Note, any unspecified parameter MUST be ignored by the receiver.

Media Type name: audio

Media subtype name: AMR

Required parameters: none

Optional parameters:
These parameters apply to RTP transfer only.

octet-align: Permissible values are 0 and 1. If 1, octet-aligned operation SHALL be used. If 0 or if not present, bandwidth efficient operation is employed.

mode-set: Requested AMR mode set. Restricts the active codec mode set to a subset of all modes. Possible values are a comma separated list of modes from the set: 0,...,7 (see Table 1a [2]). If such mode set is specified by the decoder, the encoder MUST abide by the request and MUST NOT use modes outside of the subset. If not present, all codec modes are allowed for the session.

mode-change-period: Specifies a number of frame-blocks, N, that is the interval at which codec mode changes are allowed. The initial phase of the interval is arbitrary, but

changes must be separated by multiples of N frame-blocks. If this parameter is not present, mode changes are allowed at any time during the session.

mode-change-neighbor: Permissible values are 0 and 1. If 1, mode changes SHALL only be made to the neighboring modes in the active codec mode set. Neighboring modes are the ones closest in bit rate to the current mode, either the next higher or next lower rate. If 0 or if not present, change between any two modes in the active codec mode set is allowed.

maxptime: The maximum amount of media which can be encapsulated in a payload packet, expressed as time in milliseconds. The time is calculated as the sum of the time the media present in the packet represents. The time SHOULD be a multiple of the frame size. If this parameter is not present, the sender MAY encapsulate any number of speech frames into one RTP packet.

crc: Permissible values are 0 and 1. If 1, frame CRCs SHALL be included in the payload, otherwise not. If crc=1, this also implies automatically that octet-aligned operation SHALL be used for the session.

robust-sorting: Permissible values are 0 and 1. If 1, the payload SHALL employ robust payload sorting. If 0 or if not present, simple payload sorting SHALL be used. If robust-sorting=1, this also implies automatically that octet-aligned operation SHALL be used for the session.

interleaving: Indicates that frame-block level interleaving SHALL be used for the session and its value defines the maximum number of frame-blocks allowed in an interleaving group (see Section 4.4.1). If this parameter is not present, interleaving SHALL not be used. The presence of this parameter also implies automatically that octet-aligned operation SHALL be used.

ptime: see RFC2327 [11].

channels: The number of audio channels. The possible values and their respective channel order is specified in section 4.1 in [24]. If omitted it has the default value of 1.

Encoding considerations:
This type is defined for transfer via both RTP (RFC 1889) and stored-file methods as described in Sections 4 and 5,

respectively, of RFC 3267. Audio data is binary data, and must be encoded for non-binary transport; the Base64 encoding is suitable for Email.

Security considerations:
See Section 7 of RFC 3267.

Public specification:
Please refer to Section 11 of RFC 3267.

Additional information:

The following applies to stored-file transfer methods:

Magic numbers:
single channel:
ASCII character string "#!AMR\n"
(or 0x2321414d520a in hexadecimal)
multi-channel:
ASCII character string "#!AMR_MC1.0\n"
(or 0x2321414d525F4D43312E300a in hexadecimal)

File extensions: amr, AMR
Macintosh file type code: none
Object identifier or OID: none

Person & email address to contact for further information:
johan.sjoberg@ericsson.com
ari.lakaniemi@nokia.com

Intended usage: COMMON.
It is expected that many VoIP applications (as well as mobile applications) will use this type.

Author/Change controller:
johan.sjoberg@ericsson.com
ari.lakaniemi@nokia.com
IETF Audio/Video transport working group

8.2. AMR-WB MIME Registration

The MIME subtype for the Adaptive Multi-Rate Wideband (AMR-WB) codec is allocated from the IETF tree since AMR-WB is expected to be a widely used speech codec in general VoIP applications. This MIME registration covers both real-time transfer via RTP and non-real-time transfers via stored files.

Note, any unspecified parameter MUST be ignored by the receiver.

Media Type name: audio

Media subtype name: AMR-WB

Required parameters: none

Optional parameters:

These parameters apply to RTP transfer only.

octet-align: Permissible values are 0 and 1. If 1, octet-aligned operation SHALL be used. If 0 or if not present, bandwidth efficient operation is employed.

mode-set: Requested AMR-WB mode set. Restricts the active codec mode set to a subset of all modes. Possible values are a comma separated list of modes from the set: 0,...,8 (see Table 1a [4]). If such mode set is specified by the decoder, the encoder MUST abide by the request and MUST NOT use modes outside of the subset. If not present, all codec modes are allowed for the session.

mode-change-period: Specifies a number of frame-blocks, N, that is the interval at which codec mode changes are allowed. The initial phase of the interval is arbitrary, but changes must be separated by multiples of N frame-blocks. If this parameter is not present, mode changes are allowed at any time during the session.

mode-change-neighbor: Permissible values are 0 and 1. If 1, mode changes SHALL only be made to the neighboring modes in the active codec mode set. Neighboring modes are the ones closest in bit rate to the current mode, either the next higher or next lower rate. If 0 or if not present, change between any two modes in the active codec mode set is allowed.

maxptime: The maximum amount of media which can be encapsulated in a payload packet, expressed as time in milliseconds. The time is calculated as the sum of the time the media present in the packet represents. The time SHOULD be a multiple of the frame size. If this parameter is not present, the sender MAY encapsulate any number of speech frames into one RTP packet.

crc: Permissible values are 0 and 1. If 1, frame CRCs SHALL be included in the payload, otherwise not. If crc=1, this also implies automatically that octet-aligned operation SHALL be used for the session.

robust-sorting: Permissible values are 0 and 1. If 1, the payload SHALL employ robust payload sorting. If 0 or if not present, simple payload sorting SHALL be used. If robust-sorting=1, this also implies automatically that octet-aligned operation SHALL be used for the session.

interleaving: Indicates that frame-block level interleaving SHALL be used for the session and its value defines the maximum number of frame-blocks allowed in an interleaving group (see Section 4.4.1). If this parameter is not present, interleaving SHALL not be used. The presence of this parameter also implies automatically that octet-aligned operation SHALL be used.

ptime: see RFC2327 [11].

channels: The number of audio channels. The possible values and their respective channel order is specified in section 4.1 in [24]. If omitted it has the default value of 1.

Encoding considerations:
This type is defined for transfer via both RTP (RFC 1889) and stored-file methods as described in Sections 4 and 5, respectively, of RFC 3267. Audio data is binary data, and must be encoded for non-binary transport; the Base64 encoding is suitable for Email.

Security considerations:
See Section 7 of RFC 3267.

Public specification:
Please refer to Section 11 of RFC 3267.

Additional information:
The following applies to stored-file transfer methods:

Magic numbers:
single channel:
ASCII character string "#!AMR-WB\n"
(or 0x2321414d522d57420a in hexadecimal)
multi-channel:
ASCII character string "#!AMR-WB_MC1.0\n"
(or 0x2321414d522d57425F4D43312E300a in hexadecimal)

File extensions: awb, AWB
Macintosh file type code: none
Object identifier or OID: none

Person & email address to contact for further information:
johan.sjoberg@ericsson.com
ari.lakaniemi@nokia.com

Intended usage: COMMON.
It is expected that many VoIP applications (as well as mobile applications) will use this type.

Author/Change controller:
johan.sjoberg@ericsson.com
ari.lakaniemi@nokia.com
IETF Audio/Video transport working group

8.3. Mapping MIME Parameters into SDP

The information carried in the MIME media type specification has a specific mapping to fields in the Session Description Protocol (SDP) [11], which is commonly used to describe RTP sessions. When SDP is used to specify sessions employing the AMR or AMR-WB codec, the mapping is as follows:

- The MIME type ("audio") goes in SDP "m=" as the media name.

- The MIME subtype (payload format name) goes in SDP "a=rtpmap" as the encoding name. The RTP clock rate in "a=rtpmap" MUST be 8000 for AMR and 16000 for AMR-WB, and the encoding parameters (number of channels) MUST either be explicitly set to N or omitted, implying a default value of 1. The values of N that are allowed is specified in Section 4.1 in [24].

- The parameters "ptime" and "maxptime" go in the SDP "a=ptime" and "a=maxptime" attributes, respectively.

- Any remaining parameters go in the SDP "a=fmtp" attribute by copying them directly from the MIME media type string as a semicolon separated list of parameter=value pairs.

Some example SDP session descriptions utilizing AMR and AMR-WB encodings follow. In these examples, long a=fmtp lines are folded to meet the column width constraints of this document; the backslash ("\") at the end of a line and the carriage return that follows it should be ignored.

Example of usage of AMR in a possible GSM gateway scenario:

```
m=audio 49120 RTP/AVP 97
a=rtpmap:97 AMR/8000/1
a=fmtp:97 mode-set=0,2,5,7; mode-change-period=2; \
  mode-change-neighbor=1
a=maxptime:20
```

Example of usage of AMR-WB in a possible VoIP scenario:

```
m=audio 49120 RTP/AVP 98
a=rtpmap:98 AMR-WB/16000
a=fmtp:98 octet-align=1
```

Example of usage of AMR-WB in a possible streaming scenario (two channel stereo):

```
m=audio 49120 RTP/AVP 99
a=rtpmap:99 AMR-WB/16000/2
a=fmtp:99 interleaving=30
a=maxptime:100
```

Note that the payload format (encoding) names are commonly shown in upper case. MIME subtypes are commonly shown in lower case. These names are case-insensitive in both places. Similarly, parameter names are case-insensitive both in MIME types and in the default mapping to the SDP a=fmtp attribute.

9. IANA Considerations

Two new MIME subtypes have been registered, see Section 8. A new SDP attribute "maxptime", defined in Section 8, has also been registered. The "maxptime" attribute is expected to be defined in the revision of RFC 2327 [11] and is added here with a consistent definition.

10. Acknowledgements

The authors would like to thank Petri Koskelainen, Bernhard Wimmer, Tim Fingscheidt, Sanjay Gupta, Stephen Casner, and Colin Perkins for their significant contributions made throughout the writing and reviewing of this document.

11. References

[1] 3GPP TS 26.090, "Adaptive Multi-Rate (AMR) speech transcoding", version 4.0.0 (2001-03), 3rd Generation Partnership Project (3GPP).

[2] 3GPP TS 26.101, "AMR Speech Codec Frame Structure", version 4.1.0 (2001-06), 3rd Generation Partnership Project (3GPP).

[3] 3GPP TS 26.190 "AMR Wideband speech codec; Transcoding functions", version 5.0.0 (2001-03), 3rd Generation Partnership Project (3GPP).

[4] 3GPP TS 26.201 "AMR Wideband speech codec; Frame Structure", version 5.0.0 (2001-03), 3rd Generation Partnership Project (3GPP).

[5] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

[6] 3GPP TS 26.093, "AMR Speech Codec; Source Controlled Rate operation", version 4.0.0 (2000-12), 3rd Generation Partnership Project (3GPP).

[7] 3GPP TS 26.193 "AMR Wideband Speech Codec; Source Controlled Rate operation", version 5.0.0 (2001-03), 3rd Generation Partnership Project (3GPP).

[8] Schulzrinne, H, Casner, S., Frederick, R. and V. Jacobson, "RTP: A Transport Protocol for Real-Time Applications", RFC 1889, January 1996.

[9] 3GPP TS 26.092, "AMR Speech Codec; Comfort noise aspects", version 4.0.0 (2001-03), 3rd Generation Partnership Project (3GPP).

[10] 3GPP TS 26.192 "AMR Wideband speech codec; Comfort Noise aspects", version 5.0.0 (2001-03), 3rd Generation Partnership Project (3GPP).

[11] Handley, M. and V. Jacobson, "SDP: Session Description Protocol", RFC 2327, April 1998.

[24] Schulzrinne, H., "RTP Profile for Audio and Video Conferences with Minimal Control" RFC 1890, January 1996.

11.1 Informative References

[12] GSM 06.60, "Enhanced Full Rate (EFR) speech transcoding", version 8.0.1 (2000-11), European Telecommunications Standards Institute (ETSI).

[13] ANSI/TIA/EIA-136-Rev.C, part 410 - "TDMA Cellular/PCS - Radio Interface, Enhanced Full Rate Voice Codec (ACELP)." Formerly IS-641. TIA published standard, June 1 2001.

[14] ARIB, RCR STD-27H, "Personal Digital Cellular Telecommunication System RCR Standard", Association of Radio Industries and Businesses (ARIB).

[15] Larzon, L., Degermark, M. and S. Pink, "The UDP Lite Protocol", Work in Progress.

[16] 3GPP TS 25.415 "UTRAN Iu Interface User Plane Protocols", version 4.2.0 (2001-09), 3rd Generation Partnership Project (3GPP).

[17] S. Floyd, M. Handley, J. Padhye, J. Widmer, "Equation-Based Congestion Control for Unicast Applications", ACM SIGCOMM 2000, Stockholm, Sweden .

[18] Li, A., et. al., "An RTP Payload Format for Generic FEC with Uneven Level Protection", Work in Progress.

[19] Rosenberg, J. and H. Schulzrinne, "An RTP Payload Format for Generic Forward Error Correction", RFC 2733, December 1999.

[20] 3GPP TS 26.102, "AMR speech codec interface to Iu and Uu", version 4.0.0 (2001-03), 3rd Generation Partnership Project (3GPP).

[21] 3GPP TS 26.202 "AMR Wideband speech codec; Interface to Iu and Uu", version 5.0.0 (2001-03), 3rd Generation Partnership Project (3GPP).

[22] Baugher, et. al., "The Secure Real Time Transport Protocol", Work in Progress.

[23] Perkins, C., Kouvelas, I., Hodson, O., Hardman, V., Handley, M., Bolot, J., Vega-Garcia, A. and S. Fosse-Parisis, "RTP Payload for Redundant Audio Data", RFC 2198, September 1997.

ETSI documents can be downloaded from the ETSI web server, "http://www.etsi.org/". Any 3GPP document can be downloaded from the 3GPP webserver, "http://www.3gpp.org/", see specifications. TIA documents can be obtained from "www.tiaonline.org".

12. Authors' Addresses

Johan Sjoberg
Ericsson Research
Ericsson AB
SE-164 80 Stockholm, SWEDEN

Phone: +46 8 50878230
EMail: Johan.Sjoberg@ericsson.com

Magnus Westerlund
Ericsson Research
Ericsson AB
SE-164 80 Stockholm, SWEDEN

Phone: +46 8 4048287
EMail: Magnus.Westerlund@ericsson.com

Ari Lakaniemi
Nokia Research Center
P.O.Box 407
FIN-00045 Nokia Group, FINLAND

Phone: +358-71-8008000
EMail: ari.lakaniemi@nokia.com

Qiaobing Xie
Motorola, Inc.
1501 W. Shure Drive, 2-B8
Arlington Heights, IL 60004, USA

Phone: +1-847-632-3028
EMail: qxie1@email.mot.com

13. Full Copyright Statement

Acknowledgement

Funding for the RFC Editor function is currently provided by the Internet Society.